ALGORITMOS Y ESTRUCTURAS DE DATOS III - 2$^{\underline{\text{do}}}$ Parcial

Fecha de examen: 24-JUN-2022

	Nº Orden	Apellido y nombre		L.U.	# hojas [1]
Notas:	Ej1	Ej2	Ej3	Ej4	Final

Aclaraciones: El parcial se aprueba con 2 (dos) ejercicios aprobados. Cada hoja debe estar numerada y debe tener el número de orden y L.U. El parcial dura 4 horas y es a libro abierto. Cada respuesta dada debe contar con su correspondiente **justificación** para poder ser considerada correcta.

1) Dado un digrafo D (débilmente) conexo de n vértices y m aritas y dos vértices especiales s y t, queremos calcular la mínima cantidad de aristas cuya orientación tenemos que dar vuelta para que exista un camino de s a t.

   a) Diseñar un algoritmo de complejidad $O(min\{n^2, m \log n\})$ que resuelva el problema.

   b) Demostrar que el algoritmo propuesto es correcto.

   Nota: Un digrafo es (débilmente) conexo si el grafo subyacente, que resulta de ignorar la orientación de sus aristas, es conexo.

2) Tenemos a n clientes de un supermercado $\{c_1, c_2, ..., c_n\}$ y queremos asignarle a cada uno, una caja para hacer fila. Las cajas están ordenadas en una línea y numeradas de izquierda a derecha de la 1 a la M y se encuentran separadas por pasillos. Durante el proceso de asignación algunos clientes se pelean entre sí y son separados por seguridad. Si dos clientes c_i y c_j pelean, los guardias les dicen que tienen que ponerse en filas distintas que se encuentren separadas por $K_{ij} > 0$ pasillos intermedios, para que no se vuelvan a pelear. Notar que cuando seguridad separa una pelea naturalmente hay un cliente que queda más a la izquierda (cerca de la caja 1) y el otro más a la derecha (cerca de la caja M). Con la restricción de no volver a acercarse, ese orden ya no puede cambiar. A su vez hay pares de clientes c_k y c_m que son amigos y no queremos que haya más que $L_{km} = L_{mk} \geq 0$ pasillos intermedios entre las filas de c_k y c_m. ¿será posible asignarlos a todos?

   a) Modelar el problema utilizando un sistema de resticciones de diferencias (no olviden justificar).

   b) Proponer un algoritmo polinomial que lo resuelva.

   c) ¿Qué complejidad tiene el algoritmo propuesto? Para la respuesta, tener en cuenta la cantidades m_1 y m_2 de amistades y peleas, respectivamente.

   Nota: K_{ij} de alguna manera captura la intensidad de la pelea y L_{ij} captura (inversamente) la intensidad de la amistad. Es posible que dos amigos se peleen y en ese caso hay que cumplir las dos condiciones. Si eso pasa solo puede haber soluciones si $K_{ij} \leq L_{ij}$. Para todo par de clientes sabemos si son amigos o si se pelearon, la intensidad de cada relación. Además, para aquellos clientes que se pelearon, conocemos cuál cliente quedó a la izquierda y cuál a la derecha.
   Ayuda: Si tenemos n variables x_i en un SRD y queremos acotarlas entre A y B ($x_i \in [A, B]$) podemos agregar una variable auxiliar z, sumar restricciones del tipo $A \leq x_i - z \leq B$ y luego correr la solución para que z sea 0.

[1] Incluyendo esta hoja.

3) Se tiene el mismo problema del enunciado anterior pero ahora las cajas están dispuestas en forma circular (con $M > 2$) y no podemos deducir en qué orden quedaron los clientes que fueron separados. Es decir, debemos ubicar a cada cliente de forma tal que si c_i y c_j se pelearon entonces hayan K_{ij} cajas entre c_i y c_j en ambos sentidos del círculo, sin conocer quién entre c_i y c_j será asignado a la caja de mayor número. Demostrar que este problema es NP-completo sabiendo que 3-coloreo[2] lo es.

   Ayudas:

   - Si podemos acotar las soluciones para que $x_i \in [0, 2]$ podríamos asociar los valores 0, 1 y 2 a colores.
   - Si $x_i \neq x_j$ entonces los colores asociados son distintos.

4) Un camión tiene una ruta fija de clientes $v_1, \ldots, v_n$ a la que se planificó enviarle paquetes (de alguna forma que es irrelevante). Dentro del camión se encuentra un drone que va a ser usado para visitar a $n-1$ clientes de otro conjunto $W = w_1, \ldots, w_p$ tal que $p \geq n-1$. La operación del camión y el drone es la siguiente. En cada v_i, el drone despega del camión, visita a uno de los clientes de W que aun no fue visitado, y se encuentra con el camión en el cliente v_{i+1}. El vehículo que llega primero al cliente v_{i+1} se queda esperando al vehículo más lento.

   Conociendo el tiempo t_i que demora el camión en viajar de v_i a v_{i+1} y el tiempo d_{ij} que demora el drone en viajar desde v_i hasta w_j y luego ir a v_{i+1}, queremos encontrar el camino que pueda recorrerse en el menor tiempo posible.

   a) Modelar el problema como un problema de flujo máximo con costo mínimo o, alternativamente, como uno de matching bipartito de peso mínimo (no olviden justificar).

   b) Proponer un algoritmo polinomial que resuelva el modelo propuesto.

   c) ¿Qué complejidad tiene el algoritmo propuesto? Expresar la respuesta en función de n y p, indicando la cota más ajustada que sea posible.

[2] Recordar que dado un grafo G, 3-coloreo consiste en determinar si se puede particionar a los vértices de G en a lo sumo 3 conjuntos independientes.